A Database Privacy Information Query Protocol based on Secure Multiparty Computation

Tao Zhang

Information Commission Office, Heilongjiang University, Harbin 150080, Heilongjiang, China
Zhangtao2668@126.com

Abstract

With the continuous development of information technology, database as the core part of modern information system, it undertakes the task of storage and management of data information. In the process of database query, it is of great importance to protect the security of privacy information of both parties. However, the existing programs encrypt the entire database; the encryption cost is excessively high. In order to solve this problem, the article takes the advantage of security assumptions of exchange encryption and obvious transfer technology to propose a database privacy information query program based on secure multiparty computation as well as analyzes the correctness, security, and the complexity of the design. The results show that the computational complexity of the program mentioned in the article is significantly reduced.

Keywords: *Secure multiparty computation; Database encryption; Secure query protocol*

1. Introduction

With the global development of computer and information technology, Internet provides people with quick sharing resources. The Internet information has become the most important information resources of many enterprises. Since the Internet technology is open, a lot of information in information systems, especially the sensitive information of enterprises, institutions, government departments and individuals, such as proprietary scientific research results of individuals and enterprises, customer information database of enterprises, personnel files of city fathers and other critical information resources are suffering with threat of varying degrees. Because of the sharing information resources, the event of information leak has happened in recent years. In February 2008, iPhoneUnlockUK server of British iPhone Company was hacked. The users' personal information, such as E-mail, address and other sensitive information has been copied. This event brought a very bad influence on the locals. This shows that the unsafe factors include the bug of the operating system, cyber attacks, the intentional destroy of the insiders, and even the database administrator can be a potential safety hazard. Therefore, the security of the database is a problem for military, government, business and other industries to solve. In the non-trusted environment, how to ensure the security of the database has become the focus of many scholars.

As an important research branch of secure multiparty computation, the problem of database security query has a broad application prospects in practical use. The so-called safe query means that in database query, we can ensure that the query user only gets the query results without knowing any other information in the database. At the same time, the party who has the database does not know which record the user has inquired in the database. This problem is an important application of secure multiparty computation [1] in the database,

such as the cooperation query of multiple intelligence agencies. If A wants to query a database owned by B, B's query condition is sensitive information and it cannot be leaked to B; the record database B has owned involve privacy. For the record B does not query, B should ensure that the data record is not leaked as far as possible. Therefore, how to query the database securely has become the focus of research.

Agrawal *et al.*, propose an effective program to solve the problem of secure retrieval in 2003 [2] by encrypting plaintexts of two participating sides and then encrypting each other's plaintexts for the second time. After getting the results of two sets of secondary encryption, then query its results. Without the aid of the third party it achieves the security of the database query protocol. In this program the privacy information between the two sides gets a very good protection without leaking to each other. However, the encryption calculation of the program is excessive. In 2010, Hongjia Li *et al.*, based on the concept of indiscernibility calculation and the theory of commutative encryption function [3] propose a retrieval scheme of privacy information based on any third-party. Since the third-party of the above program itself is incredible, such program caused new security risks.

For the existing problems in the above method, this paper utilizes the exchange encryption function and obvious transfer protocol, proposes an efficient security query protocol of database privacy information. This protocol needs neither the participation of obvious third-party, nor encrypting the entire database. As long as the two parties involved in the calculation can it achieve security query of relational tables in the database, and the protocol meets the security needs of the security query. The database owners do not know which record the user queried; In the meantime, the user can only get their own results but cannot get the database.

2. The Basic Concept

2.1. Secure Multiparty Computation

Secure multiparty computation [4] was first proposed by A. C. Yao in the early 1980s. It is a study in the network environment of mutual distrust. Two or more participants collaborate with each other on calculating the function of prior agreement. In the process of calculation, any participants cannot reveal their secret input information. But after the end of the calculation, it is necessary to ensure that each participant can get the results of the calculation of the function. In the past, the study of secure multiparty computation mainly focused on the theoretical research, the applied research is relatively small. However, there are a lot of applied research results of secure multiparty computation, such as privacy data retrieval, privacy protection database statistics, data mining of privacy protection, *etc*. With the promotion of secure multiparty computation applied in different areas, its application in real life is bound to become an essential part of information security system.

2.2. Exchange Encryption Function

Exchanging encryption function [5] is a combination of the encryption function. Assuming there is data v, according to the different order of combination of a pair encryption function, after the encryption v we can obtain two data $f\left(g\left(v\right)\right)$ and $g\left(f\left(v\right)\right)$. The two encrypted cipher texts meet $f\left(g\left(v\right)\right) = g\left(f\left(v\right)\right)$. If you want to restore the encrypted data, both query parties must collaborate to decrypt.

Assuming there is exchange encryption function f: $K \times M \to M$, K is the key space and M is the message space. The function f is defined in the limited domain of definition and it is computable in polynomial time.

(1) For each f_e: $M \to M$ is a bijection.

(2) For any a, $b \in K$ meets $f_a \circ f_b = f_b \circ f_a$, that is the exchangeability of function f .

2.3. Secure Multiparty Computation

Oblivious Transfer Protocol is a protocol to obtain part of the message secretly from a message, a collection. It is a very important basic protocol of secure multiparty computation. It means that the sender has two private input m. It hopes that the recipient would obtain m by probability of 50%, while the sender does not want the recipient to know whether he obtains the secret m. Since the previous OT can't meet the practical needs, thus many variants of OT has emerged later, such as reference [7-9], wherein reference [7] proposes OT_2^1, which defines as: Anna has two private input m1, m2, she wants to send one of it to Bill. She hopes that according to the choice of Bill, she is able to get one of the secret. Likewise, Bill does not want Anna to know which one he has chose. The reference [9] proposed the definition of OT_n^1, that is, at the beginning of the protocol, the sender Anna has as many as private input n, $m_1, m_2, \ldots, m_n$. She wants to send one of it to Bill. At the end of the protocol, the recipient Bill gets a certain $m_i \left(1 \le i \le n\right)$ of numerous input n, at the same time, it should be guaranteed that Bill can't get other n-1 input and Anna can't get the information i, so Anna doesn't know which one Bill has choose.

3. The Specific Programs of the Protocol

3.1. The Protocol Model

Suppose the two parties (Anna and Bill) involved in the calculation is commonly-used semi-honest participants of secure multiparty computation. The model of secure query protocol is shown in Figure 1.

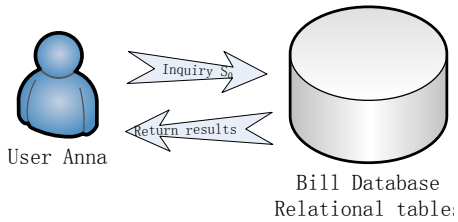

Figure 1. Security Check Protocol Model

Suppose Anna is the query user who has a query condition attribute value v, in which v is a value of the attribute V in database relational tables. Bill has a database relational table. The assumption is shown in Table 1. We assume that the value of the attribute V is set $V = \{v_1, v_2, \ldots, v_n\}$, and the corresponding record set of the attribute

value in the relational table is $\{ext(v_1), ext(v_2),\ldots, ext(v_n)\}$ .Anna wants to query whether there is v in relational table attributes V , if there is v , then return the corresponding records; if not, then the return result is empty.

Table 1. Database Rational Table

	A	V	B
$ext(v_1)$	a_1	v_1	b_1
$ext(v_2)$	a_2	v_2	b_2
…	…	…	…
$ext(v_n)$	a_n	v_n	b_n

Security query protocol:

Input: Anna inputs attribute value v . Bill inputs the database relational table.

Output: Anna gets the records she wants to query. The value of the attribute of these records is v .

Safety:

The result of the implementation of the protocol is that Anna only received the query results but not any other information in Bill's database; Bill cannot get the private query condition information v .

3.2. The Detailed Description of the Protocol

The protocol proposed in the article is based on the semi-honest model. The two participants in the communication are Anna and Bill [10]. The preparation stage: Anna selected encryption algorithm of public/private key (pk_a, sk_a) , Bill selected encryption algorithm of public key pk_a . Both parties open their public key while Anna keeps the private key sk_a secret. E ( ) is exchangeable encryption function.

The implementation process of the protocol is as follows:

Step 1：The encryption of Anna： $E_{pk_a}(v)$ ；

Anna → Bill： $E_{pk_a}(v)$

Step 2：The encryption of Bill： $E_{pk_a}\left(E_{pk_a}(v)\right)$ and

For i=1 to n{

The encryption of Bill： $E_{pk_b}(v_i)$

}

Bill →Anna： $E_{pk_b}\left(E_{pk_a}(v)\right)$, $E_{pk_b}(v_1)$, $E_{pk_b}(v_2)$, …, $E_{pk_b}(v_n)$

Step 3：The decryption of Anna：

$D_{sk_n}\left(E_{pk_b}\left(E_{pk_a}(v)\right)\right) = E_{pk_b}(v)$

For i=1 to n {

If $E_{pk_b}(v) = E_{pk_b}(v_1)$

Then $v = v_1$

Anna gets i

Else

The end of the protocol

}

Anna may get many value i, they are $\{i1, i2, .., ik\}\ (1 \le ik \le n)$. If value i does not exist, the protocol is terminated.

Step 4: Bill selected a random record number set $\{r(1), r(2), \cdots, r(n)\}$, in which the random record

$r(i) = (a_1, v_1, b_1)$, as shown in Table 3.1.

For $i = 1$ to n {

Bill calculates $rext(i) = ext(v) \oplus r(i)$:

That is $rext(i) = (ra_1, rv_1, rb_1)$

Among $ra_1 = a_1 \oplus a_1'$

$rv_1 = v_1 \oplus v_1'$

$rb_1 = b_1 \oplus b_1'$

}

Step 5: Anna and Bill executed protocol OT_n^k together twice, therein,

The first protocol OT_n^k ：

Anna input $\{i1, i2, \cdots, ik\}$, Bill input $\{rext(1).rext(2), \cdots .rext(n)\}$

After Anna and Bill executed the protocol OT_n^k,

Anna gets $\{rext(i1), rext(i2), \cdots, rext(ik)\}$;

The second protocol OT_n^k ：

Anna input $\{i1, i2, \cdots, ik\}$, Bill input $\{r(1), r(2), \cdots, r(n)\}$

After Anna and Bill executed the protocol OT_n^k,

Anna gets $\{r(i1), r(i2), \cdots, r(ik)\}$;

Step 6：for $j = i1$ to $ik\{$

Anna calculates $ext(v_1) = rext(j) \oplus r(j)$

That is $ext(v_1) = rext(j) \oplus r(j)$

Among $a_1 = ra_1 \oplus a_1'$

$v_1 = rv_1 \oplus v_1'$

$b_1 = rb_1 \oplus b_1'$

$\}$

The set $\{ext(v_1')ext(v_2'), \cdots ext(v_k')\}$ is the information the user wants to inquire.

After the execution of the protocol, Anna only gets the record information which its attribute value is v. In the meantime, Bill doesn't know which record Anna has inquired in the relational table.

4. The Performance Analysis of the Protocol

4.1. Correctness

The execution result of the protocol Theorem 4.1 is correct [11].

Prove: Encrypt the user query condition v and the attribute value v_1 of the relational table in the database through exchanging encryption function. The database sends the query conditions v and the attribute value v_1 to Anna after encryption of their own public key. Through the comparison of the cipher text value after encryption, Anna obtains the attribute value v_1 which matches the query conditions. Finally, through the execution of obvious transfer protocol twice, Anna obtains the random number and the record information after procession from the database respectively and then conducts the XOR operation between the record information and the random data and then you can get the query records. Through comparing the encrypted cipher text values, if Anna does not get the attribute value which matches the query conditions, the protocol terminates.

Thus, if there is information the user wants to query in the database, the final user is bound to get the query information he wants to query. That is, the query result of the protocol is correct.

4.2. Security

The user's privacy information of Theorem 4.2 is not leaked [11].

Prove: First, Anna encrypts the attribute value v by using the public key pk_a and then sends $E_{pk_a}(v)$ to Bill. Since the public key pk_a is open, the private key sk_a is confidential,

only Anna herself knows, Bill cannot decrypt $E_{pk_a}(v)$. Therefore, it is impossible to get any information about attribute value v. Secondly, Anna and Bill execute the protocol OT_a^k twice together. The information Anna has input twice is subscript $\{i1.i2.\cdots.ik\}$. There isn't any privacy information of Anna in subscript. Therefore, Bill cannot get any information about v.

To sum up, the user's privacy information is not leaked to the database.

Besides the information the user wants to query of theorem 4.3, other information in the database is not leaked. Prove: First, Bill encrypts $E_{pk_a}(v)$ and the value v of attribute V by using the public key pk_a and gets the result $E_{pk_b}\left(E_{pk_a}(v)\right)$ then sends it to Anna. Since the public key pk_a is open, the private key is confidential, Anna cannot decrypt. Therefore, it is impossible to get any information about v_1.

Secondly, Anna and Bill execute the protocol OT_a^k twice together. The information Bill has input is the record set $\{rext(1), rext(2), \cdots, rext(n)\}$ and random record set $\{r(1), r(2), \cdots, r(n)\}$ after XOR respectively. $rext(i)$ is the corresponding record $ext(v_1)$ of attribute value v_1 and the random record $r(i)$ after XOR. The random record set $\{r(1), r(2), \cdots, r(n)\}$ is owned privately by Bill. Therefore, after Anna executes the protocol OT_a^k twice, she can only get the record $rext(i)$ after XOR and the random record $r(i)$ she wants to know but not other record information of the relational table in the database.

To sum up, the protocol proposed in the article can guarantee the security of the database. Because besides the information the user wants to query, other information in the database is not leaked.

4.3. Complexity

The calculated amount of the protocol is mainly encryption algorithms. The letter n appeared below is the number of records of relational tables in the database.

The computational complexity: Anna executes encryption algorithms once to the attribute value v of query conditions. And she executes an encryption algorithm to $E_{pk_b}\left(E_{pk_a}(v)\right)$ send by Bill. Thus, the protocol executed encryption and decryption n +3 times.

Compare to the encryption algorithms to the entire database in reference [11], this article only encrypts the attribute value which needs query. The cost is relatively small. The following is a comparison of computational complexity of the programs shown in Table 2, where n represents the number of records; m is the number of fields in the relational tables.

Table 2. Programs Computational Complexity Comparison

Program	Computational complexity
Proposed Comparison Program	mn+3n+3
This article	n+3

Communication complexity: in protocol step 1, Anna sends the encrypted file $E_{pk_a}(v)$ to Bill, both sides need to communicate with each other; in Step 2, Bill sends Anna's $E_{pk_b}\left(E_{pk_a}(v)\right)$, $E_{pk_b}(v_1)$, $E_{pk_b}(v_2)$, …, $E_{pk_b}(v_n)$ needs to talk to each other; In step 5, in order to obtain $\{rext(i1), rext(i2), \cdots rext(ik)\}$ and random records set $\{r(i1), r(i2), \cdots r(ik)\}$, from Bill, Bill and Anna execute the protocol OT_a^k twice together. Both sides need to communicate with each other for 4 times, so the total number of mutual communication between Anna and Bill is 6 times.

5. Conclusion

This paper has designed a secure query protocol by utilizing encryption algorithms and protocol OT_a^k. The protocol has solved the security problem of the database query effectively. It didn't need the participation of the obvious third-party, the two sides involved in the calculation can achieve security query, and it has protected privacy information of both parties. The paper also has analyzed the correctness the security, and the complexity of the protocol. The results have shown that our protocol is safe and effective. As basis protocol of most secure multiparty computation, the protocol proposed in the article also carried out under a semi-honest model. Therefore, under the malicious model, how to achieve a program of higher security and lower communication cost to solve the problem of secure query better needs further study.

References

[1] O. Goldreich, S. Mieali and A. Wigderson, "How to play any mental game", Proceedings of the 19th Annual ACM Symposium on Theory of Computing, **(1987)** May 25-27, pp. 218-229.
[2] R. Agrawal, A. Evfimievski and R. Srikant, "Information Sharing Across Private Database", Proceedings of ACM SIGMOD International Conference on Management of Data, vol. 13, no. 6, **(2010)**, pp. 86-97.
[3] J. -h. Li and G. -h. Liu, "Private information retrieval program in cooperation", Computer Engineering and Design, vol. 31, no. 13, **(2010)**, pp. 2959-2961.
[4] A. C. Yao, "Protoeols for secure computations", Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, **(1982)** November 3-5, pp. 160-164.
[5] J. He, "Secret Sharing Scheme Based on Commutative Encryption Function", Computer Engineering, vol. 36, no. 9, **(2010)**, pp. 159-160.
[6] C. Crepeau, "Equivalence between two flavors of oblivious transfers", Advances in Cryptology-CRYPTO, **(1988)** August 21-25, pp. 350-354.
[7] M. Naor and B. Pinkas, "Effieient oblivious transfer Protoeols", Proceedings of the 12th annt ACM-SIAM Symposium on Diseyete Algorithms 2001, **(2001)** January 7-9, pp. 448-457.
[8] C. -K. Chu and W. G. Tzeng, "Effieient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries", Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography, **(2005)** January 11-13, pp. 172-183.
[9] W. G. Tzeng, "Effieient 1-out-of-n Oblivious Transfer Schemes with Universally Usable Parameters", IEEE Transations on Computers, vol. 52, no. 2, **(2004)**, pp. 232-240.
[10] H. Zhong, "Analysis on the key techniques of secure multiparty computation", Journal Of Anhui Agricultural University, vol. 34, no. 2, **(2007)**, pp. 291-295.
[11] W. -W. Xing and L. -S. Huang, "Design and Realization of Secure Query Scheme", Computer Engineering, vol. 32, no. 22, **(2006)**, pp. 144-146.

Author

Zhang Tao is graduated from Heilongjiang University in 2003, major in Information Management and Information Systems, and he got Bachelor Degree in Management. In the year 2008, he was admitted to Heilongjiang University, Software Engineering major as a postgraduate and got a master's degree in 2010.

In the year 2003 to March, 2011, he worked in Modern Education Technology Center of Heilongjiang University. From March 2011 till now, he works in Information Office of Heilongjiang University, Harbin, Heilongjiang Province, China.

Mr. Zhang has hosted or participated in about 20 provincial, university research projects, published a monograph and more than 20 research papers have been published in academic journals both at home and abroad, among which 3 have been retrieved by EI.